***BOOTSTRAP :***

Bootstrap est un mélange de HTML et de CSS. Les deux créateurs de Bootstrap ont créé des classes prédéfinies qu’il est possible d’utiliser pour bénéficier de leurs propriétés. Bootstrap n’est pas un nouveau langage, mais juste un nombre de classes à retenir et à réutiliser.

Pour appeler Bootstrap, il faut directement intégrer un lien dans le head qui va renvoyer vers un ***CDN (Content Delivery Network).*** Un CDN est un serveur qui va stocker des liens qui seront utilisés par beaucoup de gens. Ils sont présents partout dans le monde et c’est celui le plus proche de chez nous qui sera utilisé. De plus, beaucoup de sites les utilisent. Le cache stock des informations donc comme cache les contient déjà, ça permet de gagner en bande passante et donc en bande passante (et en argent).

Important à noter :

Il faut toujours mettre Jquery avant le Bootstrap JS dans le head et notre CSS propre à la fin.

Le contenu dans le body doit TOUJOURS être dans une div de classe ***container*** pour avoir un pourcentage de l’écran et ***container-fluid*** pour avoir 100% de l’écran. C’est cette div qui permet le responsive.

***Img-fluid*** permet d’adapter une image pour qu’elle rétrécisse en même temps que la page.

Comment faut-il penser pour coder avec Bootstrap :

Bootstrap a été pensé pour décomposer un site en 12 colonnes qui représentent chacune environs 8.30% du viewport. Il est possible de placer un élement d’une colonne x à une colonne y. La liberté est totale et il faut donc bien réfléchir à comment agencer un site pour ne pas faire n’importe quoi.

Pour que le placement soit correct et esthétique, il faut évidemment penser en terme de colonnes quand on organise le placement de ses éléments, mais ***il faut commencer par penser le placement en ligne de nos éléments.***

En effet, quand on maquette notre site, il faut savoir quel élément va se trouver sur quelle ligne. Ce sont les lignes qui seront délimitées dans les colonnes, comme ça chaque ligne peut avoir un placement différent sans être gênée par ce qui se trouve sur la ligne au-dessus ou la ligne en-dessous.

Les liens à insérer dans le head avant de commencer :

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"></script>

Il faut bien sûr veiller à utiliser les dernières versions.

Les formulaires avec Bootstrap :

Il est possible de faire pas mal de choses, comme par exemple de mettre des border rouges sur ce qui n’est pas valide. Il faut du JS pour que ça soit optimal en revanche.

<form class="form">

<div class="form-group">

<!-- Regroupe un label avec le champ utilisé -->

<label for="name"> Name:</label>

<input type="text" id="email" class="form-control">

<!-- form-control définit que le design du formulaire sera défini par Bootstrap. -->

<!-- Notre CSS venant après celui de Bootstrap, on peut apporter nos modifications à ce que propose Bootstrap. -->

<input type="button" value="envoyer" class="form-control">

<!-- Le bouton fera la même taille que les autres éléments. -->

<input type="button" value="envoyer" class="btn btn-default">

<!-- Le bouton sera plus petit. -->

<input type="button" value="envoyer" class="btn btn-primary">

<!-- Change la couleur du bouton vers une couleur préféfinie (ici bleu). Il en existe d'autres comme par exemple

warning(orange),danger(rouge) et info(gris clair). -->

</div>

<div class="form-inline"></div>

<!-- Le formulaire sera affiché sur une seule ligne le temps que c'est possible. -->

</form>

Les images et icônes :

Il est possible de modifier les images via Bootstrap. On peut les arrondir ,…

Certaines icônes sont affichées de base via des classes.

Pour les icônes de BS il faut mettre la classe dans un span.

<span class="glyphicon glyphicon-heart"></span> Coeur.

<!-- Le span doit rester vide. -->

Un autre site très utilisé s’appelle font-awesome. Si on fait appel à lui, il faut le mettre en lien dans le head. Le principe est le même mais la classe est différente et il ne faut pas utiliser un span mais un i.

<i class="far fa-hand-scissors"></i> Ciseaux.

<!-- Le i doit rester vide. -->

L’image s’adaptera si elle doit rétrécir.

<div class="col-sm-12 col-md-3 col-lg-4" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

Ici, en sm l’image prendra toute la largeur du site (12/12), en md elle prendra 3/12 donc 1/4 et en large elle en prendra 4/12. Donc 1/3.

Certaines classes permettent de modifier l’aspect esthétique des images :

<div class="col-sm-12 col-md-4 col-lg-3"><img src="goodboy.jpg" alt="Chien" class="img-fluid rounded-circle"></div>

Les tailles du responsive :

// Extra small devices (portrait phones, less than 576px)

// No media query for `xs` since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)

@media (min-width: 576px) { ... } //sm

// Medium devices (tablets, 768px and up)

@media (min-width: 768px) { ... } //md

// Large devices (desktops, 992px and up)

@media (min-width: 992px) { ... } //lg

// Extra large devices (large desktops, 1200px and up)

@media (min-width: 1200px) { ... } //xl

Il ne faut pas oublier que Bootstrap utilise énormément de divs et de classes. Par exemple, un display flex en row se traduira par :

<div class="row">

<div class="col-sm-12 col-md-4 col-lg-3" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

<div class="col-sm-12 col-md-4 col-lg-3" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

<div class="col-sm-12 col-md-4 col-lg-3" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

<div class="col-sm-12 col-md-4 col-lg-3" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

<div class="col-sm-12 col-md-4 col-lg-3" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

<div class="col-sm-12 col-md-4 col-lg-3" ><img src="goodboy.jpg" alt="Chien" class="img-fluid"></div>

</div>

Ici, la div de classe row donne la propriété display flex et flex-flow row à toutes les div contenues à l’intérieur.

***Bootstrap est « mobile first ».*** ***Si on ne précise pas de taille (ex : col-3), il sera automatiquement adapté au format mobile !***

Le offset :

Il est possible de placer une div à partir d’une certaine colonne sur la ligne. Par exemple, une div basique non placée via CSS sera automatiquement placée sur la colonne 1. On peut par exemple décider qu’elle démarre sur la colonne 4.

<div class="col-md-4 offset-4">Div offset-4</div>

On peut définir une colonne à partir de laquelle démarrer selon la taille de l’écran :

div class="offset-md-4 col-md-4 offset-sm-1 col-sm-8">Div offset-4</div>

<!-- Ici, quand l'écran sera en md ou supérieur, alors la div commencera sur la 4ème colonne

et fera une taille de 4 colonnes.

Quand l'écran est petit, alors la div partira de la première colonne et fera 8 des 12 colonnes. -->

Il faut savoir que l’écran s’adaptera toujours à la taille maximum prévue. Par exemple, si on définit les tailles jusqu’en md, même le lg et le xl auront l’apparence du md au niveau des réglages.

<div class="container">

<div class="row">

<div class="offset-sm-1 col-sm-10 offset-md-3 col-md-6 offset-lg-0 col-lg-12 black">1</div>

<div class="offset-sm-1 col-sm-2 offset-md-2 col-md-3 offset-lg-0 col-lg-1 black">2</div>

<div class="offset-sm-6 col-sm-2 offset-md-2 col-md-3 offset-lg-10 col-lg-1 black">3</div>

<div class="offset-sm-1 col-sm-10 offset-md-1 col-md-10 offset-lg-3 col-lg-6 black">4</div>

<div class="offset-sm-4 col-sm-2 offset-md-3 col-md-2 offset-lg-1 col-lg-5 black">5</div>

<div class="offset-sm-0 col-sm-2 offset-md-2 col-md-2 offset-lg-0 col-lg-5 black">6</div>

</div>

</div>

Dans une div row, les div se mettront automatiquement à la ligne si elles dépassent la douzième colonne. Sur deux div étant sur une même ligne, le offset de la deuxième ne part pas de zéro mais de la div d’avant. Par exemple ,si on a une div placée en colonne 2/3 , et qu’on veut placer une div en colonne 4, alors le offset de la deuxième div sera offset-0.

Le offset-0 est important et il faut le noter. Si on veut changer la valeur du offset, il suffit juste de changer la valeur.

Les display :

Le display peut être géré en responsive avec Bootstrap. La classe qui gère le display est d-taille.

<div class="container">

<div class="text-danger d-lg-block d-md-none d-sm-none d-none" value="1">

<h1>Noooon, la fenêtre est trop grande!</h1>

</div>

<div class="text-success d-lg-none d-md-block d-sm-none d-none">

<h2>Ah! La fenêtre est à la bonne taille!</h2>

</div>

<div class="text-warning d-lg-none d-md-none d-sm-block d-none">

<h3>Oups, la fenêtre est trop petite!</h3>

</div>

</div>

Les Navbars :

Il faut impérativement ajouter

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>

Bootstrap 4 ne gère plus la position du navbar donc il faut mettre un autre DNS qui le gère à la place.

Ici, un menu cliquable classique :

<div class="container ">

<div class="row">

<div class="dropdown col-2 offset-1">

<button class="btn dropdown-toggle" type="button" data-toggle="dropdown">

Equipe

</button>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Alexia Dupont</a></li>

<li><a class="dropdown-item" href="#">Martine Laposte</a></li>

<li><a class="dropdown-item" href="#">Marc Pierre</a></li>

<li><a class="dropdown-item"

</ul>

</div>

Ici, le dropdown-divider permet de séparer les items comme le ferait un <hr> :

<div class="dropdown col-2 offset-1">

<button class="btn dropdown-toggle" type="button" data-toggle="dropdown">

Equipe</span>

</button>

<ul class="dropdown-menu">

<li><a class="dropdown-item" href="#">Alexia Dupont</a></li>

<li><a class="dropdown-item" href="#">Martine Laposte</a></li>

<li class="dropdown-divider"></li>

<li><a class="dropdown-item" href="#">Marc Pierre</a></li>

<li><a class="dropdown-item" href="#">René lajoie</a></li>

</ul>

</div>

Enfin, ici les différentes sections ont un titre non cliquable (le dropdown-header) :

<div class="dropdown col-2 offset-1">

<button class="btn dropdown-toggle" type="button" data-toggle="dropdown">

Equipe</span>

</button>

<ul class="dropdown-menu">

<li class="dropdown-header"><span class="text-info ">Service Marketing</span></li>

<li><a class="dropdown-item" href="#">Alexia Dupont</a></li>

<li><a class="dropdown-item" href="#">Martine Laposte</a></li>

<li class="dropdown-header"><span class="text-info ">Service Paie</span></li>

<li><a class="dropdown-item" href="#">Marc Pierre</a></li>

<li><a class="dropdown-item" href="#">René lajoie</a></li>

</ul>

</div>

Les Navbars qui toggle en responsive :

Il est possible grâce à Bootstrap de faire le fameux menu hamburger qui toggle au rétrécissement de la page (en général au format mobile), ou tout simplement faire un menu qui toggle :

<nav class="navbar navbar-expand-sm bg-info navbar-light">

<!-- Le navbar-expand permet de passer une navbar horizontale à la verticale à une certaine -->

<!-- taille. C'est grâce à ça que les navbars mobile sont en verticale. -->

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">A propos</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Projets</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Compétences</a>

</li>

</ul>

</nav>

Ici, la navbar est classique et passera juste en vertical sans se transformer en hamburger.

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

<!-- C'est à partir du sm que la barre va expand et faire apparaître le hamburger -->

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#mymenu">

<!-- Le data-target sert à cibler un ID qui apparaîtra au clic. Le data-toggle="collapse" est

important également.Il va déclencher le bouton au collapse de la navbar. -->

<span class="navbar-toggler-icon"></span>Menu

<!-- C'est ce span qui contient l'icône des trois traits. Elle est gérée par Bootstrap.

C'est nécécessairement un span. On peut écrire quelque -->

</button>

<div class="collapse navbar-collapse" id="mymenu">

<!-- Cette div avec la classe collapse navbar-collapse et l'ID du data-target contient tout ce qui sera dans le hamburger.

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="#">A propos</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Projets</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Compétences</a>

</li>

</ul>

</div>

</nav>

Un exemple concret :

<nav class="navbar navbar-expand-sm bg-dark navbar-dark">

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#mymenu">

<span class="navbar-toggler-icon"></span> Le menu est là!

</button>

<div class="collapse navbar-collapse" id="mymenu">

<ul class="navbar-nav">

<li class="nav-item">

<a href="#" class="nav-link">Choix 1</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link">Choix 2</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link">Choix 3</a>

</li>

<li class="nav-item">

<a href="#" class="nav-link">Choix 4</a>

</li>

</ul>

</div>

</nav>